

Amendment to the Claims

This listing of claims will replace all prior versions and listing of claims in the application:

Listing of Claims:

1. (currently amended) A method of protecting a selected region of an image from subsequent editing, the method comprising the steps of:

(a) creating a first texture comprising a plurality of pixels each with an assigned scalar value indicating a level of protection for a corresponding pixel of a protected image;

(b) following a first user signal, directing graphical input representing a plurality of brush strokes performed by the user into a second texture, wherein the protected image is at least initially unedited by the graphical input and wherein the plurality of brush strokes comprises at least one overlapping portion;

(c) modifying a value of at least one pixel of the second texture using the first texture;
and

(d) following a second user signal subsequent to the first user signal, blending at least one pixel of the second texture into the protected image; and

(e) displaying an image based on the result of step (d).

2. – 3. (canceled)

4. (currently amended) The method of claim 13, wherein the at least one overlapping portion corresponds to an area overlapped by a plurality of brush strokes.

5. (currently amended) The method of claim 12, wherein the plurality of brush strokes comprises at least one member selected from the group consisting of a paint stroke, an erase stroke, a pencil stroke, a pen stroke, a line application, a character application, a text application, a batch deletion, a batch paste, and a flood fill.
6. (original) The method of claim 5, further comprising the step of generating the graphical input of step (b), wherein the graphical input corresponds to a movement of a user.
7. (original) The method of claim 6, wherein the step of generating the graphical input comprises assigning scalar values to pixels of a scratch texture that correspond to a transition region at one or more edges of a brush stroke.
8. (original) The method of claim 6, wherein the graphical input comprises a scratch texture representing a brush stroke and step (b) comprises blending the scratch texture into the second texture substantially upon completion of the brush stroke.
9. (original) The method of claim 8, wherein the step of blending the scratch texture into the second texture comprises performing a compositing operation.
10. (original) The method of claim 9, wherein the compositing operation is an overlay operation performed with pixels of A and B, where A comprises pixels having a paint color attenuated by the scratch texture and B comprises the second texture.

11. (original) The method of claim 6, wherein the step of generating the graphical input comprises the step of, for each of at least a plurality of pixels of a scratch texture:

(i) comparing a candidate scalar value from received data to an existing scalar value at a corresponding pixel of the scratch texture; and

(ii) assigning the candidate scalar value to the corresponding pixel of the scratch texture only if the candidate scalar value exceeds the existing scalar value.

12. (original) The method of claim 1, wherein the graphical input represents at least one brush stroke performed by a user, and wherein the at least one brush stroke comprises at least one overlapping portion corresponding to an area of a single brush stroke that overlaps itself.

13. (currently amended) The method of claim 1, further comprising the steps of:

(i)~~(e)~~ copying at least one pixel of the protected image into a display image; and

(ii)~~(f)~~ blending at least one pixel of the second texture into the display image.

14. (currently amended) The method of claim 13, wherein step (i)~~(e)~~ and step (ii)~~(f)~~ are each performed prior to step (d).

15. (currently amended) The method of claim 13, wherein at least one of step (c) and step (ii)~~(f)~~ proceeds pixel-by-pixel as the second texture accumulates graphical input.

16. (currently amended) The method of claim 13, wherein step (i)~~(e)~~ and step (ii)~~(f)~~ are each performed prior to step (d), and wherein at least one of step (c) and step (ii)~~(f)~~ proceeds pixel-by-

pixel as the second stencil texture accumulates graphical input.

17. (original) The method of claim 1, wherein the graphical input represents at least one erase stroke performed by a user.

18. (currently amended) The method of claim 17, further comprising the step of:

~~(i)~~(e) modifying a value of at least one pixel in the protected image using the first texture.

19. (currently amended) The method of claim 18, wherein step ~~(i)~~(e) comprises attenuating a value of a pixel in the protected image subject to a minimum RGB α alpha value, where the minimum alpha value is determined using the first texture.

20. (original) The method of claim 1, wherein the graphical input represents at least one paint stroke and at least one erase stroke performed by a user.

21. (original) The method of claim 1, wherein step (c) comprises attenuating values of pixels of the second texture using values of corresponding pixels in the first texture.

22. (original) The method of claim 1, wherein step (d) comprises performing a compositing operation.

23. (original) The method of claim 22, wherein the compositing operation is an overlay

operation performed with pixels of A and B, where A comprises the second texture and B comprises the protected image.

24. (original) The method of claim 23, wherein A comprises the second texture as modified in step (c).

25. (original) The method of claim 1, wherein step (c) and step (d) are performed substantially simultaneously.

26. (original) The method of claim 1, wherein the assigned scalar value of a pixel in the first texture indicates a level of protection from 0% to 100%.

27. (original) The method of claim 26, wherein the level of protection is a nonzero value less than 100%.

28. (original) The method of claim 26, wherein the level of protection relates to an opacity.

29. (original) The method of claim 1, wherein the protected image is unedited by the graphical input of step (b) until the blending in step (d).

30. – 31. (canceled)

32. (currently amended) The method of claim ~~131~~, wherein the graphical input in step (b)

represents a plurality of paint strokes performed by the user between the first user signal and the second user signal.

33. (currently amended) The method of claim ~~130~~, wherein the first user signal is a button click.

34. (original) The method of claim 1, wherein the first texture represents at least one user-selected region of the image.

35. – 45. (canceled)

46. (new) A method of protecting a selected region of an image from subsequent editing, the method comprising the steps of:

- (a) creating a first texture comprising a plurality of pixels each with an assigned scalar value indicating a level of protection for a corresponding pixel of a protected image;
- (b) directing graphical input comprising at least one erase stroke into a second texture, wherein the protected image is at least initially unedited by the graphical input;
- (c) modifying a value of at least one pixel of the second texture using the first texture;
- (d) attenuating a value of a pixel in the protected image subject to a minimum RGBa alpha value determined from the first texture;
- (e) blending at least one pixel of the second texture into the protected image; and
- (f) displaying an image based on the result of step (e).

47. (new) The method of claim 46, wherein the graphical input further comprises at least one member selected from the group consisting of a paint stroke, a pencil stroke, a pen stroke, a line application, a character application, a text application, a batch deletion, a batch paste, and a flood fill.

48. (new) The method of claim 46, wherein the graphical input comprises a plurality of erase strokes comprising at least one overlapping portion.

49. (new) The method of claim 46, wherein the assigned scalar value of a pixel in the first texture indicates a level of protection from 0% to 100%.

50. (new) The method of claim 49, wherein the level of protection is a nonzero value less than 100%.

51. (new) The method of claim 49, wherein the level of protection relates to an opacity.

52. (new) An apparatus for protecting a selected region of an image from subsequent editing, the apparatus comprising:

- (a) a graphical user interface device; and
- (b) a processor configured to run software that:
 - (i) creates a first texture comprising a plurality of pixels each with an assigned scalar value indicating a level of protection for a corresponding pixel of a protected image;

(ii) following a first user signal, directs graphical input from the graphical user interface device into a second texture, wherein the graphical input represents a plurality of brush strokes performed by the user, wherein the protected image is at least initially unedited by the graphical input, and wherein the plurality of brush strokes comprises at least one overlapping portion;

(iii) modifies a value of at least one pixel of the second texture using the first texture;

(iv) following a second user signal subsequent to the first user signal, blends at least one pixel of the second texture into the protected image; and

(v) displays an image based on the result of (iv).

53. (new) The apparatus of claim 52, wherein the plurality of brush strokes comprises at least one member selected from the group consisting of a paint stroke, an erase stroke, a pencil stroke, a pen stroke, a line application, a character application, a text application, a batch deletion, a batch paste, and a flood fill.

54. (new) The apparatus of claim 52, wherein the assigned scalar value of a pixel in the first texture indicates a level of protection from 0% to 100%.

55. (new) The apparatus of claim 54, wherein the level of protection is a nonzero value less than 100%.

56. (new) The apparatus of claim 54, wherein the level of protection relates to an opacity.

57. (new) The apparatus of claim 52, wherein the protected image is unedited by the graphical input of (ii) until the blending in (iv).